Redox Reactions and Proton Pumps

Peter Gawthrop (peter.gawthrop@unimelb.edu au)

September 7, 2020

Contents

Note: The example is discussed in detail by (Gawthrop and Pan, 2020) available here and should be read in conjunction.

Note: this is the Redox.ipynb notebook. The PDF version "Redox Reactions and Proton Pumps" is available here.

1 Introduction

Redox reactions and proton pumps play a fundamental role in Biology. This note illustrates this using a bond graph model of complex I of the mitochondrial electron transport chain.

```
In [1]: ## Some useful imports
        import BondGraphTools as bgt
        import numpy as np
        import sympy as sp
        import matplotlib.pyplot as plt

        ## Stoichiometric analysis
        import stoich as st

        ## SVG
        import svgBondGraph as sbg

        ## Display (eg disp.SVG(), disp.
        import IPython.display as disp

        ## Potential data
        import phiData

        quiet = True
```

```
In /home/peterg/.local/lib/python3.6/site-packages/matplotlib/mpl-data/stylelib/_classic_test.mp
The text.latex.unicode rcparam was deprecated in Matplotlib 3.0 and will be removed in 3.2.
In /home/peterg/.local/lib/python3.6/site-packages/matplotlib/mpl-data/stylelib/_classic_test.mp
The savefig.frameon rcparam was deprecated in Matplotlib 3.1 and will be removed in 3.3.
In /home/peterg/.local/lib/python3.6/site-packages/matplotlib/mpl-data/stylelib/_classic_test.mp
The pgf.debug rcparam was deprecated in Matplotlib 3.0 and will be removed in 3.2.
In /home/peterg/.local/lib/python3.6/site-packages/matplotlib/mpl-data/stylelib/_classic_test.mp
The verbose.level rcparam was deprecated in Matplotlib 3.1 and will be removed in 3.3.
In /home/peterg/.local/lib/python3.6/site-packages/matplotlib/mpl-data/stylelib/_classic_test.mp
The verbose.fileo rcparam was deprecated in Matplotlib 3.1 and will be removed in 3.3.
```

2 Redox reaction

A key energy-generating redox reaction that within the mitochondrial respiratory chain is

$$\mathrm{NADH} + \mathrm{Q} + \mathrm{H}^{+} \overset{\mathrm{r}}{\rightleftharpoons} \mathrm{NAD}^{+} + \mathrm{QH}_2 \tag{1}$$

This reaction can be divided into the half reactions:

$$\mathrm{NADH} \overset{r_1}{\rightleftharpoons} 2\,e_1^- + H^+ + \mathrm{NAD}^+ \tag{2}$$

$$2\,e_2^- + 2\,H^+ + Q \overset{r_2}{\rightleftharpoons} QH_2 \tag{3}$$

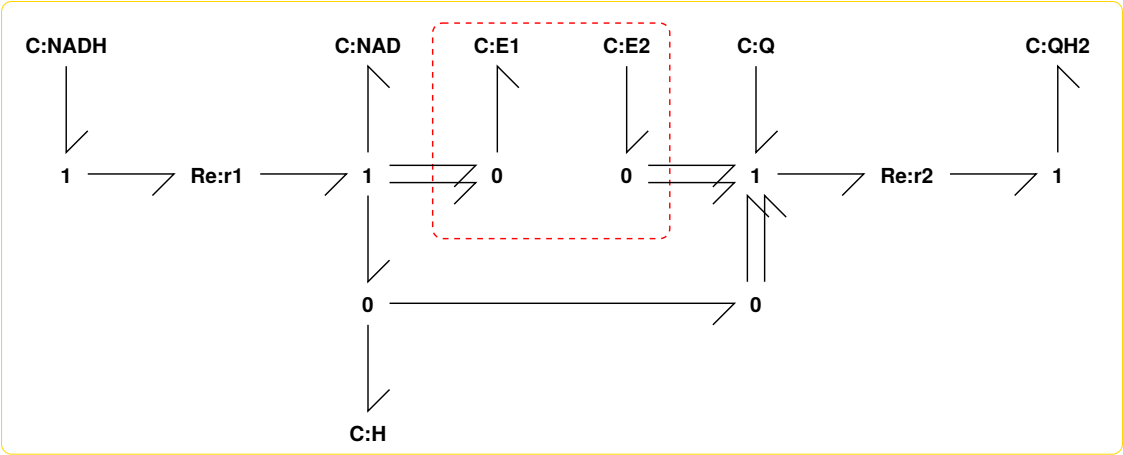
A bond graph representation of this decomposition is given below.

2.1 Bond graph

C:E1 and C:E2 represent the electron potentials and the other The C components represent the species; the two Re components the two half reactions.

```
In [2]: ## Redox reaction
        sbg.model('Redox_abg.svg')
        import Redox_abg
        disp.SVG('Redox_abg.svg')
```

Out[2]:



2.2 Stoichiometry

```
In [3]: ## Stoichiometry
        linear = ['E1','E2']
        s = st.stoich(Redox_abg.model(),linear=linear,quiet=quiet)
        chemostats = ['NADH','NAD','Q','QH2','H']
        sc = st.statify(s,chemostats=chemostats)

In [4]: ## Stoichiometric matrix
        disp.Latex(st.sprintl(s,'N'))
```

Out[4]:

$$N = \begin{pmatrix} 2 & 0 \\ 0 & -2 \\ 1 & -2 \\ 1 & 0 \\ -1 & 0 \\ 0 & -1 \\ 0 & 1 \end{pmatrix} \tag{4}$$

2.3 Reactions

These are automatically generated from the bond graph

```
In [5]: ## Reactions
        disp.Latex(st.sprintrl(s,chemformula=True,all=True))
```

Out[5]:

$$\mathrm{NADH} \overset{r_1}{\rightleftharpoons} 2\,\mathrm{E_1} + \mathrm{H} + \mathrm{NAD} \tag{5}$$

$$2\,\mathrm{E_2} + 2\,\mathrm{H} + \mathrm{Q} \overset{r_2}{\rightleftharpoons} \mathrm{QH_2} \tag{6}$$

2.4 Potentials

The reaction (Faraday Equivalent) potentials are computed from tables gleaned from the literature

```
In [6]: ## Standard potetials
        phi_Std = phiData.phi_Std()

        ## Typical concentrations
        conc = phiData.ParRubXu16_conc()

        ## From BazBeaVin16
        conc['Q'] = conc['QH2'] = 1e-2

        ## pH 7
        conc['H'] = 1e-7

        ## Table for paper and put values in to phi_NADH etc.
        print('%% Table')
        ch='\ch'
        l='{'
        r='}'
        eol = r'\\'
        phi_std = {}
        for spec in ['NAD','NADH','Q','QH2','H']:
            phi0 = phi_Std[spec]
```

```
            con = conc[spec]
            phi_std_spec = phi0 + st.V_N()*np.log(con)
            phi_std[spec] = phi_std_spec
            #print(f'phi_Std_{spec} = {1000*phi0:0.0f}, phi_{spec} = {1000*phi_std[spec]:.0f}, c
            print(f'{ch}{l}{spec}{r} & {1000*phi0:.0f} & {con:1.2e} & {1000*phi_std_spec:.0f}{eo
            exec(f'phi_{spec} = {phi_std_spec}')


        ## Print the worked example for the paper.
        print('\n%% Equations')
        E1 = 0.5*(phi_NADH - phi_NAD - phi_H)
        E2 = 0.5*(phi_QH2 - phi_Q - 2*phi_H)
        print(f'E1 = 0.5({1000*phi_NADH:.0f} - {1000*phi_NAD:.0f} - {1000*phi_H:.0f}) = {1000*E1
        print(f'E2 = 0.5({1000*phi_QH2:.0f} - {1000*phi_Q:.0f} - 2x{1000*phi_H:.0f}) = {1000*E2:
        print(f'E1-E2 = {1000*(E1-E2):.0f} mV')
        print(f'PMF = {1000*(E1-E2)/2:.0f} mV')
```

```
%% Table
\ch{NAD} & 188 & 5.02e-04 & -15\\
\ch{NADH} & 407 & 7.50e-05 & 154\\
\ch{Q} & 675 & 1.00e-02 & 552\\
\ch{QH2} & -241 & 1.00e-02 & -365\\
\ch{H} & 0 & 1.00e-07 & -431\\

%% Equations
E1 = 0.5(154 - -15 - -431) = 300 mV
E2 = 0.5(-365 - 552 - 2x-431) = -28 mV
E1-E2 = 328 mV
PMF = 164 mV
```

3 Proton pump

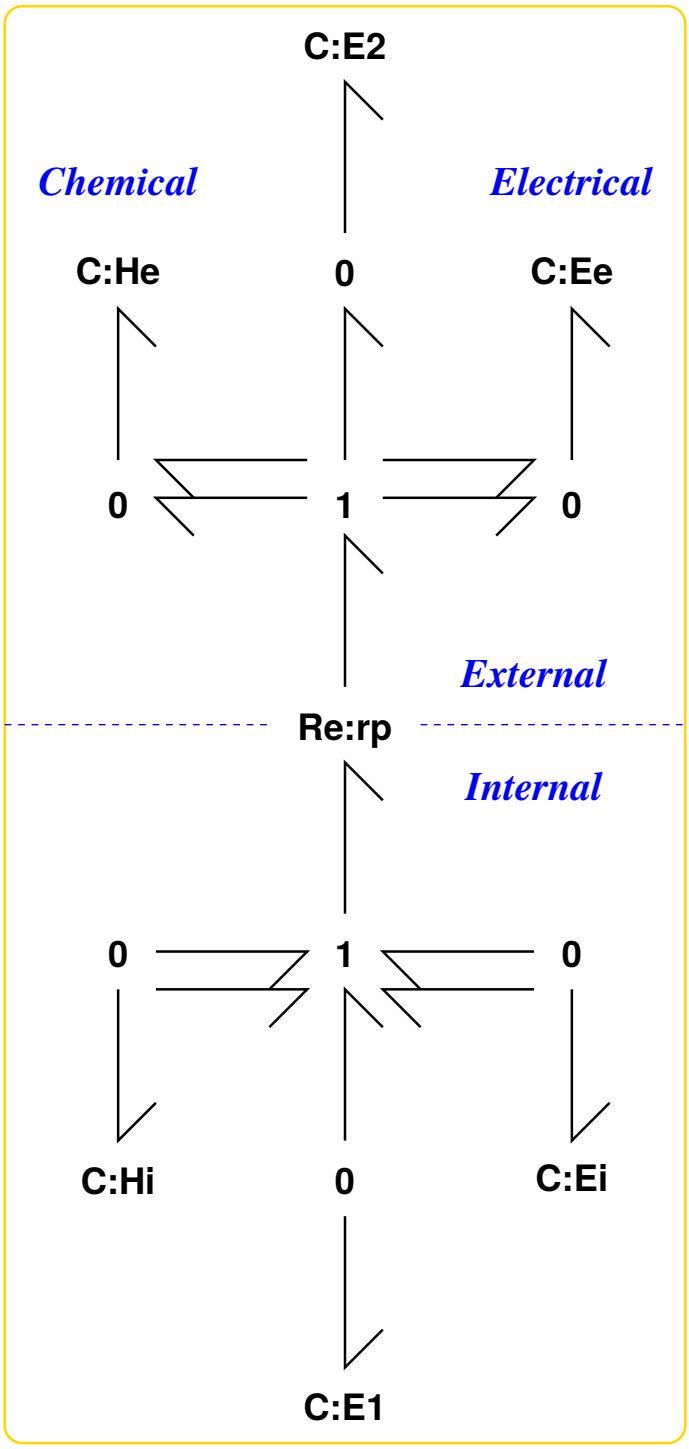
The redox reaction of complex I drives a proton pump.

3.1 Bond graph

C:E1 and C:E2 correspond to the redox reaction and provide the potential to drive protons in the interior H_i^+ to the exterior H_e^+ of the mitochondrial membrane. The protons have both electrical and chemoical potential.

```
In [7]: ## Proton pump
        sbg.model('ProtonPump_abg.svg')
        import ProtonPump_abg
        disp.SVG('ProtonPump_abg.svg')
```

```
   Out[7]:
```



3.2 Stoichiometry

```
In [8]: ## Stoichiometry
        linear = ['E1','E2','Ei','Ee']
```

```
s = st.stoich(ProtonPump_abg.model(),linear=linear,quiet=quiet)
chemostats = ['E1','E2','Ei','Ee','Hi','He']
sc = st.statify(s,chemostats=chemostats)
#print(s['species'])
#disp.Latex(st.sprint(s0,'K'))
#print(st.sprints(s))
```

3.3 Reactions

These are automatically generated from the bond graph

In [9]:
```
## Reactions
disp.Latex(st.sprintrl(s,chemformula=True,all=True))
```

Out[9]:

$$\mathrm{E_1} + 2\,\mathrm{Ei} + 2\,\mathrm{Hi} \overset{\mathrm{rp}}{\rightleftharpoons} \mathrm{E_2} + 2\,\mathrm{Ee} + 2\,\mathrm{He} \tag{7}$$

In [10]:
```
## Flows
disp.Latex(st.sprintvl(s))
```

Out[10]:

$$v_{rp} = \kappa_{rp}\left(-K_{He}^2 x_{He}^2 e^{\frac{K_{E2}x_{E2}+2K_{Ee}x_{Ee}}{V_N}} + K_{Hi}^2 x_{Hi}^2 e^{\frac{K_{E1}x_{E1}+2K_{Ei}x_{Ei}}{V_N}}\right) \tag{8}$$

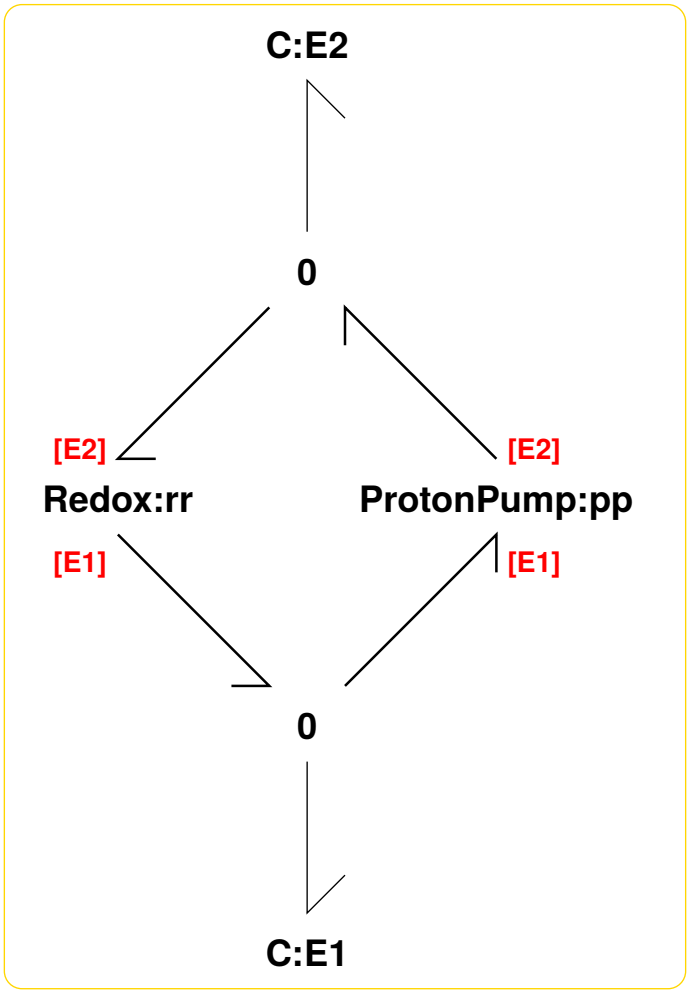
4 Complex I

The model of mitochondrial complex I comprides tow modules: the redox reaction and the proton pump.

In [11]:
```
## Complex I
sbg.model('ComplexI_abg.svg')
import ComplexI_abg
disp.SVG('ComplexI_abg.svg')
```

```
Creating subsystem: ProtonPump:pp
Creating subsystem: Redox:rr
```

Out[11]:



4.1 Stoichiometry

```
In [12]: ## Stoichiometry
         linear = ['E1','E2','pp_Ei','pp_Ee']
         s = st.stoich(ComplexI_abg.model(),linear=linear,quiet=quiet)
         print(s['species'])
         chemostats = ['pp_Ee', 'pp_Ei', 'pp_He', 'pp_Hi', 'rr_H', 'rr_NAD', 'rr_NADH', 'rr_Q',
         sc = st.statify(s,chemostats=chemostats)
         #print(s['species'])
         #disp.Latex(st.sprint(s0,'K'))
         #print(st.sprints(s))
```

```
['E1', 'E2', 'pp_Ee', 'pp_Ei', 'pp_He', 'pp_Hi', 'rr_H', 'rr_NAD', 'rr_NADH', 'rr_Q', 'rr_QH2']
```

4.2 Reactions

These are automatically generated from the bond graph

```
In [13]: ## Reactions
         disp.Latex(st.sprintrl(s,chemformula=True,all=True))
```

Out[13]:

$$E_1 + 2\,pp_Ei + 2\,pp_Hi \xrightleftharpoons{pp_rp} E_2 + 2\,pp_Ee + 2\,pp_He \tag{9}$$

$$rr_NADH \xrightleftharpoons{rr} 2\,E_1 + rr_H + rr_NAD \tag{10}$$

$$2\,E_2 + 2\,rr_H + rr_Q \xrightleftharpoons{rr} rr_QH_2 \tag{11}$$

```
In [14]: ## Flows
         disp.Latex(st.sprintvl(s))
```

Out[14]:

$$v_{pprp} = \kappa_{pprp}\left(-K_{ppHe}^2 x_{ppHe}^2 e^{\frac{K_{E2}x_{E2}+2K_{ppEe}x_{ppEe}}{V_N}} + K_{ppHi}^2 x_{ppHi}^2 e^{\frac{K_{E1}x_{E1}+2K_{ppEi}x_{ppEi}}{V_N}}\right) \tag{12}$$

$$v_{rr} = \kappa_{rr}\left(-K_{rrH}K_{rrNAD}x_{rrH}x_{rrNAD}e^{\frac{2K_{E1}x_{E1}}{V_N}} + K_{rrNADH}x_{rrNADH}\right) \tag{13}$$

$$v_{rr} = \kappa_{rr}\left(K_{rrH}^2 K_{rrQ}x_{rrH}^2 x_{rrQ}e^{\frac{2K_{E2}x_{E2}}{V_N}} - K_{rrQH2}x_{rrQH2}\right) \tag{14}$$

```
In [15]: ## Path
         sp = st.path(s,sc)
         ## Reactions
         disp.Latex(st.sprintrl(sp,chemformula=True,all=True))
```

Out[15]:

$$4\,pp_Ei + 4\,pp_Hi + 2\,rr_H + rr_NADH + rr_Q \xrightleftharpoons{pr_1} 4\,pp_Ee + 4\,pp_He + rr_H + rr_NAD + rr_QH_2 \tag{15}$$

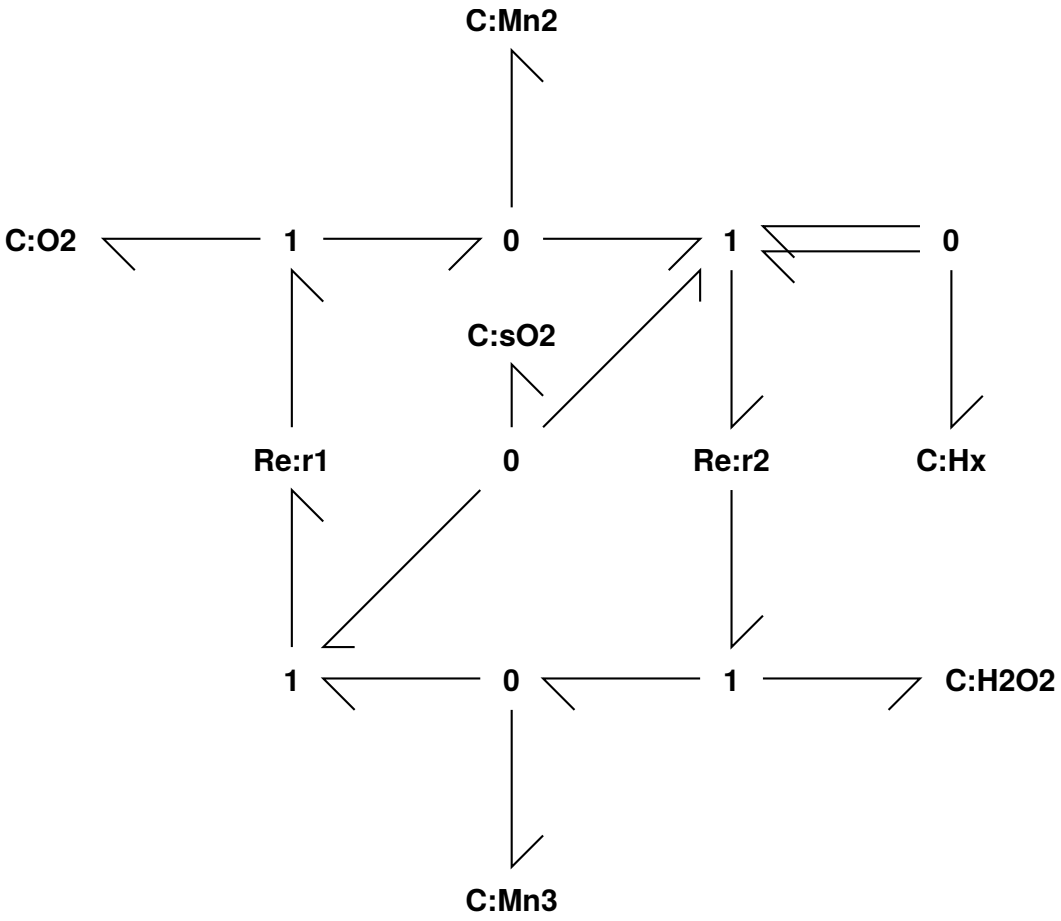
5 Superoxide generation and removal (additional material)

Superoxide is generated from the electron potential of the redox reaction and free oxygen by the reaction: $e^- + O_2 \rightleftharpoons O_2^{\bullet -}$ It is removed by superoxide dismutase in conjunction with glutathione peroxidase. THe bond graph representation of these mechanisms are given below.

5.1 Superoxide dismutase (MnSOD)

```
In [16]: ## Complex I
         sbg.model('MnSOD_abg.svg')
         import MnSOD_abg
         disp.SVG('MnSOD_abg.svg')
```

Out[16]:




```
In [17]: ## Stoichiometry
         s = st.stoich(MnSOD_abg.model(),quiet=quiet)
         chemostats = ['H2O2', 'Hx', 'O2', 'sO2']
         sc = st.statify(s,chemostats=chemostats)
         #disp.Latex(st.sprint(s0,'K'))
         #print(st.sprints(s))
```

```
In [18]: ## Reactions
         disp.Latex(st.sprintrl(s,chemformula=True,all=True))
```

Out[18]:

$$\mathrm{Mn_3} + \mathrm{sO_2} \overset{\mathrm{r_1}}{\rightleftharpoons} \mathrm{Mn_2} + \mathrm{O_2} \tag{16}$$

$$2\,\mathrm{Hx} + \mathrm{Mn_2} + \mathrm{sO_2} \overset{\mathrm{r_2}}{\rightleftharpoons} \mathrm{H_2O_2} + \mathrm{Mn_3} \tag{17}$$

```
In [19]: ## Flows
         disp.Latex(st.sprintvl(s))
```

Out[19]:

$$v_{r1} = \kappa_{r1}\left(-K_{Mn2}K_{O2}x_{Mn2}x_{O2} + K_{Mn3}K_{sO2}x_{Mn3}x_{sO2}\right) \tag{18}$$

$$v_{r2} = \kappa_{r2}\left(-K_{H2O2}K_{Mn3}x_{H2O2}x_{Mn3} + K_{Hx}^2K_{Mn2}K_{sO2}x_{Hx}^2x_{Mn2}x_{sO2}\right) \tag{19}$$

In [20]:
```
sp = st.path(s,sc)
## Reactions
disp.Latex(st.sprintrl(sp,chemformula=True,all=False))
```

Out[20]:

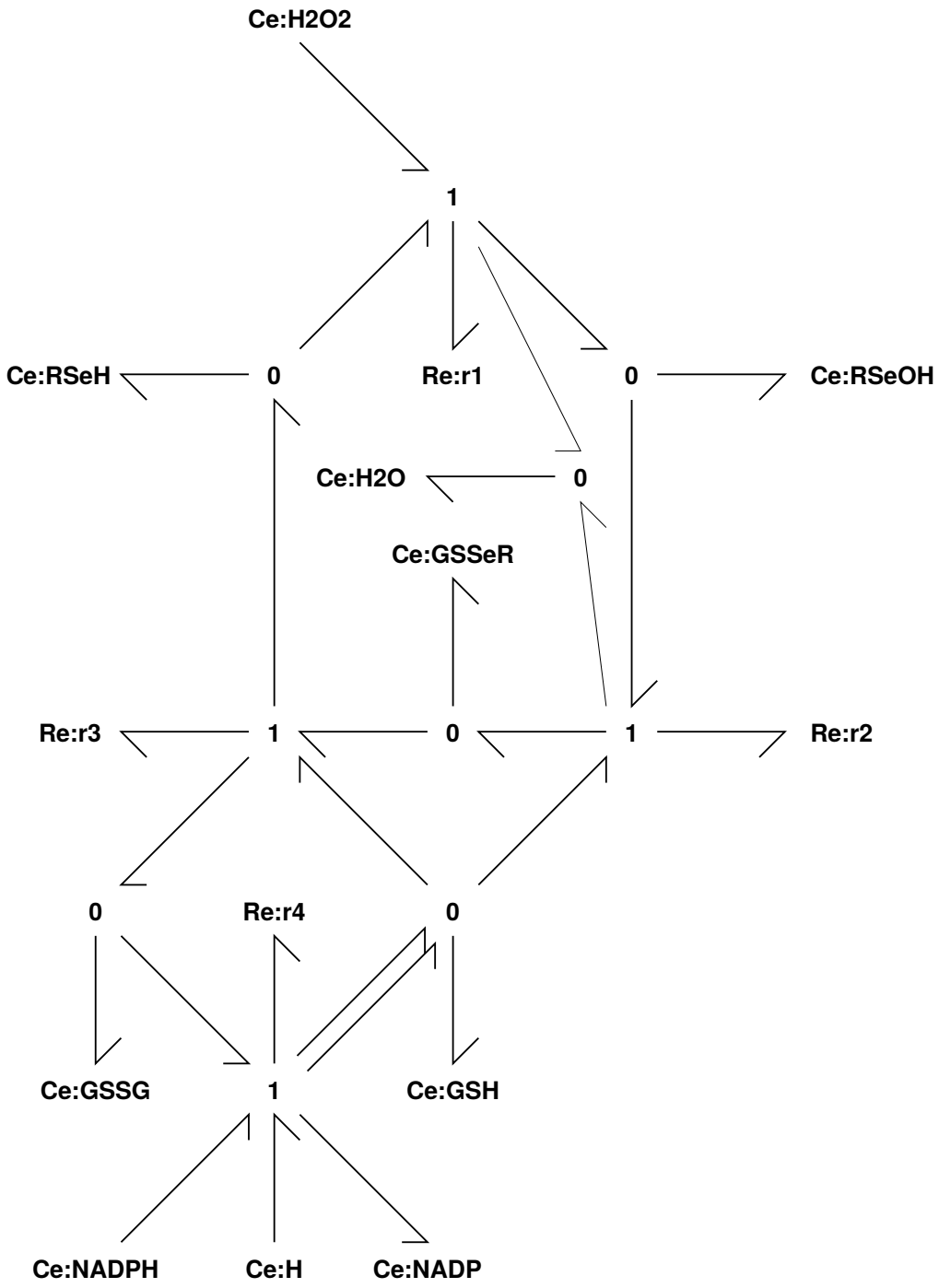
$$2\,\mathrm{Hx} + 2\,\mathrm{sO_2} \xrightleftharpoons{\mathrm{pr_1}} \mathrm{H_2O_2} + \mathrm{O_2} \tag{20}$$

5.2 Glutathione peroxidase (GPx1)

In [21]:
```
## Complex I
sbg.model('GPx1_abg.svg')
import GPx1_abg
disp.SVG('GPx1_abg.svg')
```

```
Converting one-port r1 to two-port
Converting one-port r4 to two-port
Converting one-port r2 to two-port
Converting one-port r3 to two-port
```

Out[21]:




```
In [22]: ## Stoichiometry
         s = st.stoich(GPx1_abg.model(),quiet=quiet)
         #print(s['species'])
         chemostats = ['GSH', 'GSSG', 'H', 'H2O', 'H2O2','NADP','NADPH']
         sc = st.statify(s,chemostats=chemostats)
```

```
        #disp.Latex(st.sprint(s0,'K'))
        #print(st.sprints(s))
```

In [23]:
```
## Reactions
disp.Latex(st.sprintrl(s,chemformula=True,all=True))
```

Out[23]:

$$H_2O_2 + RSeH \overset{r_1}{\rightleftharpoons} H_2O + RSeOH \tag{21}$$
$$GSH + RSeOH \overset{r_2}{\rightleftharpoons} GSSeR + H_2O \tag{22}$$
$$GSH + GSSeR \overset{r_3}{\rightleftharpoons} GSSG + RSeH \tag{23}$$
$$GSSG + H + NADPH \overset{r_4}{\rightleftharpoons} 2\,GSH + NADP \tag{24}$$

In [24]:
```
## Flows
disp.Latex(st.sprintvl(s))
```

Out[24]:

$$v_{r1} = \kappa_{r1}\left(-K_{H2O}K_{RSeOH}x_{H2O}x_{RSeOH} + K_{H2O2}K_{RSeH}x_{H2O2}x_{RSeH}\right) \tag{25}$$
$$v_{r2} = \kappa_{r2}\left(K_{GSH}K_{RSeOH}x_{GSH}x_{RSeOH} - K_{GSSeR}K_{H2O}x_{GSSeR}x_{H2O}\right) \tag{26}$$
$$v_{r3} = \kappa_{r3}\left(K_{GSH}K_{GSSeR}x_{GSH}x_{GSSeR} - K_{GSSG}K_{RSeH}x_{GSSG}x_{RSeH}\right) \tag{27}$$
$$v_{r4} = \kappa_{r4}\left(-K_{GSH}^2K_{NADP}x_{GSH}^2x_{NADP} + K_{GSSG}K_HK_{NADPH}x_{GSSG}x_Hx_{NADPH}\right) \tag{28}$$

In [25]:
```
sp = st.path(s,sc)
## Reactions
disp.Latex(st.sprintrl(sp,chemformula=True,all=True))
```

Out[25]:

$$2\,GSH + H_2O_2 + RSeOH \overset{pr_1}{\rightleftharpoons} GSSG + 2\,H_2O + RSeOH \tag{29}$$

References

Peter J. Gawthrop and Michael Pan. Network thermodynamical modelling of bioelectrical systems: A bond graph approach. Available at arXiv:2009.02217, 2020.